UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/656,096	09/05/2003	Hideomi Idei	16869S-094000US	9922

20350 7590 06/26/2008

TOWNSEND AND TOWNSEND AND CREW, LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

EXAMINER	
GILLIS, BRIAN J	

ART UNIT	PAPER NUMBER
2141	

MAIL DATE	DELIVERY MODE
06/26/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/656,096	IDEI ET AL.	
	Examiner	Art Unit	
	Brian J. Gillis	2141	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11 March 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,4-8,10-14,16-20,22,24 and 26* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,4-8,10-14,16-20,22,24 and 26* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *05 September 2003 and 18 August 2006* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Claims 1, 2, 4, 7, 8, 10, 13, 14, and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shillo (US PGPUB US2003/0110263) in view of Naik et al (US PGPUB US2004/0205206) in view of Huntington et al (US PGPUB US2003/0131098) in view of Kurihara et al (US Patent #6,867,872) in view of Hazelwood et al (Non Patent Literature).

Claim 1 discloses a management server connected to a plurality of servers to manage storage areas includes in storage apparatuses as virtual storage areas; wherein said storage apparatuses are shared by said plurality of servers; said storage

apparatuses include assignment areas which are storage areas assigned to at least one of said plurality of servers; data stored in said assignment areas of said storage apparatuses includes high-priority data having high priority and low-priority data having low priority; said management server judges whether data to be written in said storage apparatuses is the high-priority data of the low priority data on the basis of a write request of data from one of said plurality of servers and keeps a judgment result and position information of storage areas in which said data is written; and said management server being responsive to an area assignment instruction of storage areas exceeding unassigned areas received from one of said plurality of servers to release at least part of said assignment areas of other servers as unassigned areas and assign released areas to one of said plurality of servers, wherein upon receiving an area assignment instruction, the management server judges whether (i) a size of the unassigned areas exceeds a size of the storage areas specified by said area assignment instruction, (ii) a total size of the unassigned areas and unused areas exceeds the size of the storage areas specified by said area assignment instruction, or (iii) a total size of the unassigned areas, the unused areas and storage areas having stored low-priority data exceeds the size of the storage areas specified by said area assignment instruction, and when the condition (iii) is met, said management server releases at least part of storage areas in which the low-priority data is stored, of the assignment areas of other servers as unassigned areas and assigns at least areas to one of said plurality of servers; wherein when condition (iii) is met, said management server releases storage areas and assignment areas, said management server:

determines a storage area having a largest number of blocks in which low-priority data is stored, the storage area being of the assignment areas of other servers in which low-priority data is stored, wherein the blocks in which low-priority data is stored comprise contiguous blocks, non-contiguous blocks or a combination of non-contiguous and contiguous blocks of low-priority data; releases as unassigned area the storage area having the largest number of blocks in which low-priority data is stored; after releasing the storage area, determines whether the total size of the unassigned areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction; wherein if the total size of the total size of the unassigned areas and the unused areas does not exceed the size of the storage areas specified by said area assignment instruction, said management server iteratively performs the steps of determining a storage area, releasing a storage area, and determining whether the total size of the unassigned, the unused areas and the released storage area exceeds the size of the storage areas specified by said area assignment instruction until the total size of the total size of the unassigned areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction or no additional storages areas in which low priority data is stored are available. Shillo teaches storage areas are shared by multiple servers (paragraph 41), the virtual storage pool made by the grouping of the storage resources knows how much space each application is allocated in the pool (paragraph 42), and a re-allocation process takes place to re-allocate unused resources which are assigned to applications (paragraph 43). It fails to teach data stored in said assignment areas of said storage apparatuses includes high-

priority data having high priority and low-priority data having low priority, the server judges whether data to be written in said storage apparatuses is the high-priority data or the low-priority data on the basis of a write request of data from one of said plurality of servers and keeps a judgment result and position information of storage areas in which said data is written, and wherein upon receiving an area assignment instruction, the management server judges whether (i) a size of the unassigned areas exceeds a size of the storage areas specified by said area assignment instruction, (ii) a total size of the unassigned areas and unused areas exceeds the size of the storage areas specified by said area assignment instruction, or (iii) a total size of the unassigned areas, the unused areas and storage areas having stored low-priority data exceeds the size of the storage areas specified by said area assignment instruction, and when the condition (iii) is met, said management server releases at least part of storage areas in which the low-priority data is stored, of the assignment areas of other servers as unassigned areas and assigns at least areas to one of said plurality of servers wherein when condition (iii) is met, said management server releases storage areas and assignment areas, said management server: determines a storage area having a largest number of blocks in which low-priority data is stored, the storage area being of the assignment areas of other servers in which low-priority data is stored, wherein the blocks in which low-priority data is stored comprise contiguous blocks, non-contiguous blocks or a combination of non-contiguous and contiguous blocks of low-priority data releases as unassigned area the storage area having the largest number of blocks in which low-priority data is stored; after releasing the storage area, determines whether the total size of the unassigned

areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction wherein if the total size of the total size of the unassigned areas and the unused areas does not exceed the size of the storage areas specified by said area assignment instruction, said management server iteratively performs the steps of determining a storage area, releasing a storage area, and determining whether the total size of the unassigned, the unused areas and the released storage area exceeds the size of the storage areas specified by said area assignment instruction until the total size of the total size of the unassigned areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction or no additional storages areas in which low priority data is stored are available. Naik et al teaches a server assigns priority based on the user who issued the request and keeps information on the mapping (paragraphs 63, 64, and 69).

Shillo and Naik et al are analogous art because they are both related to managing storage on a network.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to use the priority flagging and judging in Naik et al with the system in Shillo because inefficient use of available resources and high costs are avoided (Naik, paragraph 71).

Shillo in view of Naik et al teaches the limitations as recited above. It fails to teach upon receiving an area assignment instruction, the management server judges whether (i) a size of the unassigned areas exceeds a size of the storage areas specified by said area assignment instruction, (ii) a total size of the unassigned areas and unused

areas exceeds the size of the storage areas specified by said area assignment instruction, or (iii) a total size of the unassigned areas, the unused areas and storage areas having stored low-priority data exceeds the size of the storage areas specified by said area assignment instruction, and when the condition (iii) is met, said management server releases at least part of storage areas in which the low-priority data is stored, of the assignment areas of other servers as unassigned areas and assigns at least areas to one of said plurality of servers wherein when condition (iii) is met, said management server releases storage areas and assignment areas, said management server: determines a storage area having a largest number of blocks in which low-priority data is stored, the storage area being of the assignment areas of other servers in which low-priority data is stored, wherein the blocks in which low-priority data is stored comprise contiguous blocks, non-contiguous blocks or a combination of non-contiguous and contiguous blocks of low-priority data; releases as unassigned area the storage area having the largest number of blocks in which low-priority data is stored; after releasing the storage area, determines whether the total size of the unassigned areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction; wherein if the total size of the total size of the unassigned areas and the unused areas does not exceed the size of the storage areas specified by said area assignment instruction, said management server iteratively performs the steps of determining a storage area, releasing a storage area, and determining whether the total size of the unassigned, the unused areas and the released storage area exceeds the size of the storage areas specified by said area assignment instruction until the total

size of the total size of the unassigned areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction or no additional storages areas in which low priority data is stored are available. Huntington et al teaches data is distinguished by priority (paragraph 62) and if free space is needed for a request for storage the server may recycle sections that have low priority data within (figure 5, paragraphs 62, 63, and 69).

Shillo in view of Naik et al and Huntington et al are analogous art because they are both related to managing network storage resources.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to use the recycling of low priority data storage in Huntington et al with the system in Shillo in view of Naik et al because large numbers of packets may be communicated and stored on a network with minimal user intervention (Huntington, paragraph 5).

Shillo in view of Naik et al in view of Huntington et al teaches the limitations as recited above. It fails to teach the management server determines a storage area having a largest number of blocks in which low-priority data is stored, the storage area being of the assignment areas of other servers in which low-priority data is stored, wherein the blocks in which low-priority data is stored comprise contiguous blocks, non-contiguous blocks or a combination of non-contiguous and contiguous blocks of low-priority data; releases as unassigned area the storage area having the largest number of blocks in which low-priority data is stored; after releasing the storage area, determines whether the total size of the unassigned areas and the unused areas

exceeds the size of the storage areas specified by said area assignment instruction; wherein if the total size of the total size of the unassigned areas and the unused areas does not exceed the size of the storage areas specified by said area assignment instruction, said management server iteratively performs the steps of determining a storage area, releasing a storage area, and determining whether the total size of the unassigned, the unused areas and the released storage area exceeds the size of the storage areas specified by said area assignment instruction until the total size of the total size of the unassigned areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction or no additional storages areas in which low priority data is stored are available. Kurihara et al teaches releasing data of low priority in the cache (column 19, lines 5-19), determining if the cache has free space (column 19, lines 5-19), and releasing data is repeated until it is determined enough space is available (column 19, lines 5-19).

Shillo in view of Naik et al in view of Huntington et al and Kurihara et al are analogous art because they are both related to managing network data storing.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to use the low priority data releasing feature in Kurihara et al with the system in Shillo in view of Naik et al in view of Huntington et al because efficient use of storage areas is provided (Kurihara, column 4, lines 51-58).

Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al teach the limitations as recited above. It fails to teach determining a storage area having a largest number of blocks in which low-priority data is stored, wherein the

blocks in which low-priority data is stored comprise contiguous blocks, non-contiguous blocks or a combination of non-contiguous and contiguous blocks of low-priority data. Hazelwood et al teaches the largest block contiguous or non contiguous memory currently available is freed first (section 4.5).

Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al and Hazelwood et al are analogous art because they are both related to managing network data storing.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to use the largest block first release feature in Hazelwood et al with the system in Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al because operation execution speed is increased (Hazelwood, section 1, paragraph 1).

Claim 2 discloses a management server according to claim 1, wherein said assignment areas of said storage apparatuses include used areas and unused areas; and said management server includes information for identifying said used areas and said unused areas of said assignment areas; said management server being responsive to an area assignment instruction of storage areas exceeding the unassigned areas received from one of said plurality of servers to release at least part of said unused areas of said assignment areas of other servers on the basis of said identification information as unassigned areas and assign released areas to one of said plurality of servers. Shillo further teaches the virtual storage pool has used and unused areas (paragraph 43), a server can detect how much allocated space each application actually

uses (paragraph 42), and the managing server reallocates the unused portion of the allocated space (paragraph 43).

Claim 4 discloses a management server according to claim 2, wherein data stored in the used areas in said assignment areas of said storage apparatuses includes high-priority data having high priority and low-priority data having low priority; and said management server judges whether data to be written in said storage apparatuses is the high-priority data or the low-priority data on the basis of a write request of data from said server and keeps judgment result and position information of storage areas in which said data is written; said management server being responsive to an area assignment instruction of storage areas exceeding the unassigned areas received from one of said plurality of servers to release at least part of unused areas and at least part of areas in which the low-priority data is stored, of the assignment areas of other servers as unassigned areas and assign released areas to one of said plurality of servers. Shillo further teaches data stored in assigned areas of devices, and reallocating unused resources already allocated (paragraphs 42 and 43). Huntington et al further teaches data is distinguished by priority (paragraph 62) and if free space is needed for a request for storage the server may recycle sections that have low priority data within (figure 5, paragraphs 62, 63, and 69).

Claim 7 discloses a storage apparatus system comprising: a storage apparatuses; and a management server connected to a plurality of servers and said storage apparatuses; said management server managing storage areas of said storage apparatuses as virtual storage areas; said storage apparatuses being shared by said

plurality of servers; said storage apparatuses including assignment areas which are storage areas assigned to at least one of said plurality of servers; data stored in said assignment areas of said storage apparatuses includes high-priority data having high priority and low-priority data having low priority; said management server judges whether data to be written in said storage apparatuses is the high-priority data or the low-priority data on the basis of a write request of data from one of said plurality of servers and keeps judgment result and position information of storage areas in which said data is written; said management server being responsive to an area assignment instruction of storage areas exceeding unassigned areas received from one of said plurality of servers to release at least one of assignment areas of other servers as unassigned area and assign released areas to one of said plurality of servers, wherein upon receiving an area assignment instruction, the management server judges whether (i) a size of the unassigned areas exceeds a size of the storage areas specified by said area assignment instruction, (ii) a total size of the unassigned areas and unused areas exceeds the size of the storage areas specified by said area assignment instruction, or (iii) a total size of the unassigned areas, the unused areas and storage areas having stored low-priority data exceeds the size of the storage areas specified by said area assignment instruction, and when condition (iii) is met, said management server releases at least part of storage areas in which low-priority data is stored, or the assignment areas of other servers as unassigned areas and assigns at least areas to one of said plurality of servers; wherein when condition (iii) is met, said management server releases storage areas and assignment areas, said management server:

determines a storage area having a largest number of blocks in which low-priority data is stored, the storage area being of the assignment areas of other servers in which low-priority data is stored, wherein the blocks in which low-priority data is stored comprise contiguous blocks, non-contiguous blocks or a combination of non-contiguous and contiguous blocks of low-priority data; releases as unassigned area the storage area having the largest number of blocks in which low-priority data is stored; after releasing the storage area, determines whether the total size of the unassigned areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction; wherein if the total size of the total size of the unassigned areas and the unused areas does not exceed the size of the storage areas specified by said area assignment instruction, said management server iteratively performs the steps of determining a storage area, releasing a storage area, and determining whether the total size of the unassigned, the unused areas and the released storage area exceeds the size of the storage areas specified by said area assignment instruction until the total size of the total size of the unassigned areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction or no additional storages areas in which low priority data is stored are available. Shillo teaches a managing server and multiple storage devices (paragraphs 41 and 42), the managing server manages a virtual storage pool which is a collection of all the storage resources available (paragraph 42), storage areas are shared by multiple servers (paragraph 41), the virtual storage pool made by the grouping of the storage resources knows how much space each application is allocated in the pool (paragraph 42), and a re-allocation

process takes place to re-allocate unused resources which are assigned to applications (paragraph 43). It fails to teach of data stored in said assignment areas of said storage apparatuses includes high-priority data having high priority and low-priority data having low priority, the server judges whether data to be written in said storage apparatuses is the high-priority data or the low-priority data on the basis of a write request of data from one of said plurality of servers and keeps a judgment result and position information of storage areas in which said data is written, and wherein upon receiving an area assignment instruction, the management server judges whether (i) a size of the unassigned areas exceeds a size of the storage areas specified by said area assignment instruction, (ii) a total size of the unassigned areas and unused areas exceeds the size of the storage areas specified by said area assignment instruction, or (iii) a total size of the unassigned areas, the unused areas and storage areas having stored low-priority data exceeds the size of the storage areas specified by said area assignment instruction, and when the condition (iii) is met, said management server releases at least part of storage areas in which the low-priority data is stored, of the assignment areas of other servers as unassigned areas and assigns at least areas to one of said plurality of servers wherein when condition (iii) is met, said management server releases storage areas and assignment areas, said management server: determines a storage area having a largest number of blocks in which low-priority data is stored, the storage area being of the assignment areas of other servers in which low-priority data is stored, wherein the blocks in which low-priority data is stored comprise contiguous blocks, non-contiguous blocks or a combination of non-contiguous and

contiguous blocks of low-priority data releases as unassigned area the storage area having the largest number of blocks in which low-priority data is stored; after releasing the storage area, determines whether the total size of the unassigned areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction wherein if the total size of the total size of the unassigned areas and the unused areas does not exceed the size of the storage areas specified by said area assignment instruction, said management server iteratively performs the steps of determining a storage area, releasing a storage area, and determining whether the total size of the unassigned, the unused areas and the released storage area exceeds the size of the storage areas specified by said area assignment instruction until the total size of the total size of the unassigned areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction or no additional storages areas in which low priority data is stored are available. Naik et al teaches a server assigns priority based on the user who issued the request and keeps information on the mapping (paragraphs 63, 64, and 69).

Shillo and Naik et al are analogous art because they are both related to managing storage on a network.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to use the priority flagging and judging in Naik et al with the system in Shillo because inefficient use of available resources and high costs are avoided (Naik, paragraph 71).

Shillo in view of Naik et al teaches the limitations as recited above. It fails to teach upon receiving an area assignment instruction, the management server judges whether (i) a size of the unassigned areas exceeds a size of the storage areas specified by said area assignment instruction, (ii) a total size of the unassigned areas and unused areas exceeds the size of the storage areas specified by said area assignment instruction, or (iii) a total size of the unassigned areas, the unused areas and storage areas having stored low-priority data exceeds the size of the storage areas specified by said area assignment instruction, and when the condition (iii) is met, said management server releases at least part of storage areas in which the low-priority data is stored, of the assignment areas of other servers as unassigned areas and assigns at least areas to one of said plurality of servers wherein when condition (iii) is met, said management server releases storage areas and assignment areas, said management server: determines a storage area having a largest number of blocks in which low-priority data is stored, the storage area being of the assignment areas of other servers in which low-priority data is stored, wherein the blocks in which low-priority data is stored comprise contiguous blocks, non-contiguous blocks or a combination of non-contiguous and contiguous blocks of low-priority data, releases as unassigned area the storage area having the largest number of blocks in which low-priority data is stored; after releasing the storage area, determines whether the total size of the unassigned areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction wherein if the total size of the total size of the unassigned areas and the unused areas does not exceed the size of the storage areas specified by said area

assignment instruction, said management server iteratively performs the steps of determining a storage area, releasing a storage area, and determining whether the total size of the unassigned, the unused areas and the released storage area exceeds the size of the storage areas specified by said area assignment instruction until the total size of the total size of the unassigned areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction or no additional storages areas in which low priority data is stored are available. Huntington et al teaches data is distinguished by priority (paragraph 62) and if free space is needed for a request for storage the server may recycle sections that have low priority data within (figure 5, paragraphs 62, 63, and 69).

Shillo in view of Naik et al and Huntington et al are analogous art because they are both related to managing network storage resources.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to use the recycling of low priority data storage in Huntington et al with the system in Shillo in view of Naik et al because large numbers of packets may be communicated and stored on a network with minimal user intervention (Huntington, paragraph 5).

Shillo in view of Naik et al in view of Huntington et al teaches the limitations of as recited above. It fails to teach the management server determines a storage area having a largest number of blocks in which low-priority data is stored, the storage area being of the assignment areas of other servers in which low-priority data is stored wherein the blocks in which low-priority data is stored comprise contiguous blocks, non-

contiguous blocks or a combination of non-contiguous and contiguous blocks of low-priority data; releases as unassigned area the storage area having the largest number of blocks in which low-priority data is stored; after releasing the storage area, determines whether the total size of the unassigned areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction; wherein if the total size of the total size of the unassigned areas and the unused areas does not exceed the size of the storage areas specified by said area assignment instruction, said management server iteratively performs the steps of determining a storage area, releasing a storage area, and determining whether the total size of the unassigned, the unused areas and the released storage area exceeds the size of the storage areas specified by said area assignment instruction until the total size of the total size of the unassigned areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction or no additional storages areas in which low priority data is stored are available. Kurihara et al teaches releasing data of low priority in the cache (column 19, lines 5-19), determining if the cache has free space (column 19, lines 5-19), and releasing data is repeated until it is determined enough space is available (column 19, lines 5-19).

Shillo in view of Naik et al in view of Huntington et al and Kurihara et al are analogous art because they are both related to managing network data storing.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to use the low priority data releasing feature in Kurihara et al with the

system in Shillo in view of Naik et al in view of Huntington et al because efficient use of storage areas is provided (Kurihara, column 4, lines 51-58).

Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al teach the limitations as recited above. It fails to teach determining a storage area having a largest number of blocks in which low-priority data is stored wherein the blocks in which low-priority data is stored comprise contiguous blocks, non-contiguous blocks or a combination of non-contiguous and contiguous blocks of low-priority data. Hazelwood et al teaches the largest block contiguous or non contiguous memory currently available is freed first (section 4.5).

Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al and Hazelwood et al are analogous art because they are both related to managing network data storing.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to use the largest block first release feature in Hazelwood et al with the system in Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al because operation execution speed is increased (Hazelwood, section 1, paragraph 1).

Claim 8 discloses a storage apparatus system according to claim 7, wherein said assignment areas of said storage apparatuses include used areas and unused areas; and said management server includes information for identifying said used areas and said unused areas of said assignment areas; said management server being responsive to an area assignment instruction of storage areas exceeding the unassigned areas received from one of said plurality of servers to release at least part of said unused

areas of other servers on the basis of said identification information as unassigned areas and assign released areas to one of said plurality of servers. Shillo further teaches the virtual storage pool has used and unused areas (paragraph 43), a server can detect how much allocated space each application actually uses (paragraph 42), and the managing server reallocates the unused portion of the allocated space (paragraph 43).

Claim 10 discloses a storage apparatus system according to claim 8, wherein data stored in said used areas of said storage apparatuses includes high-priority data having high priority and low-priority data having low priority; and said management server judges whether data to be written in said storage apparatuses is the high-priority data or the low-priority data on the basis of a write request of data from one of said plurality of servers and keeps judgment result and position information of storage areas in which said data is written; said management server being responsive to an area assignment instruction of storage areas exceeding the unassigned areas received from one of said plurality of servers to release at least part of said unused areas and at least part of areas in which the low-priority data is stored, of the assignment areas of other servers as unassigned areas and assign the areas to one of said plurality of servers. Shillo further teaches data stored in assigned areas of devices, and reallocating unused resources already allocated (paragraphs 42 and 43). Huntington et al further teaches data is distinguished by priority (paragraph 62) and if free space is needed for a request for storage the server may recycle sections that have low priority data within (figure 5, paragraphs 62, 63, and 69).

Claim 13 discloses a computer program product for a management server which manages storage areas included in storage apparatuses as virtual storage areas, wherein said management server is connected to a plurality of servers; and said storage apparatuses are shared by said plurality of servers through said management server and include assignment areas which are storage areas assigned to at least one of said plurality of servers, wherein data stored in said assignment areas of said storage apparatuses include high-priority data having high priority and low-priority data having low priority; said computer program product comprising: a code for judging on the basis of a write request of data from one of said plurality of servers whether data to be written in said storage apparatuses is said high-priority data or said low-priority data; and a code for information indicative of judgment result and position of storage areas in which said data is written; and code for being responsive to an area assignment instruction of storage areas exceeding unassigned areas received from one of said plurality of servers to release at least part of assignment areas of other servers as unassigned areas and assign released area to one of said plurality of servers, wherein upon receiving an area assignment instruction, the code for being responsive to an area assignment instruction judges whether (i) a size of the unassigned areas exceeds a size of the storage areas specified by said area assignment instruction, (ii) a total size of the unassigned areas and unused areas exceeds the size of the storage areas specified by said area assignment instruction, or (iii) a total size of the unassigned areas, the unused areas and storage areas having stored low-priority data exceeds the size of the storage areas specified by said area assignment instruction, and when the condition (iii) is met,

said management server releases at least part of storage areas in which the low-priority data is stored, of the assignment areas of other servers as unassigned areas and assigns at least areas to one of said plurality of servers a code executed when condition (iii) is met, the code including: a code for determining a storage area having a largest number of blocks in which low-priority data is stored, the storage area being of the assignment areas of other servers in which low-priority data is stored wherein the blocks in which low-priority data is stored comprise contiguous blocks, non-contiguous blocks or a combination of non-contiguous and contiguous blocks of low-priority data; a code for releasing as unassigned area the storage area having the largest number of blocks in which low-priority data is stored; a code for determining whether the total size of the unassigned areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction after releasing as unassigned area the storage area having the largest number of blocks in which low priority data was stored; wherein if the total size of the total size of the unassigned areas and the unused areas does not exceed the size of the storage areas specified by said area assignment instruction, said management server iteratively performs the steps of determining a storage area, releasing a storage area, and determining whether the total size of the unassigned, the unused areas and the released storage area exceeds the size of the storage areas specified by said area assignment instruction until the total size of the total size of the unassigned areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction or no additional storages areas in which low priority data is stored are available; and a computer readable storage medium for storing said

code. Shillo teaches a re-allocation process takes place to re-allocate unused resources which are assigned to applications (paragraph 43), and a computer program product on a computer-readable medium (page 6 #14). It fails to teach of data stored in said assignment areas of said storage apparatuses includes high-priority data having high priority and low-priority data having low priority, the server judges whether data to be written in said storage apparatuses is the high-priority data or the low-priority data on the basis of a write request of data from one of said plurality of servers and keeps a judgment result and position information of storage areas in which said data is written, and wherein upon receiving an area assignment instruction, the management server judges whether (i) a size of the unassigned areas exceeds a size of the storage areas specified by said area assignment instruction, (ii) a total size of the unassigned areas and unused areas exceeds the size of the storage areas specified by said area assignment instruction, or (iii) a total size of the unassigned areas, the unused areas and storage areas having stored low-priority data exceeds the size of the storage areas specified by said area assignment instruction, and when the condition (iii) is met, said management server releases at least part of storage areas in which the low-priority data is stored, of the assignment areas of other servers as unassigned areas and assigns at least areas to one of said plurality of servers a code executed when condition (iii) is met, the code including: a code for determining a storage area having a largest number of blocks in which low-priority data is stored, the storage area being of the assignment areas of other servers in which low-priority data is stored wherein the blocks in which low-priority data is stored comprise contiguous blocks, non-contiguous blocks or a

combination of non-contiguous and contiguous blocks of low-priority data, a code for releasing as unassigned area the storage area having the largest number of blocks in which low-priority data is stored, a code for determining whether the total size of the unassigned areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction after releasing as unassigned area the storage area having the largest number of blocks in which low priority data was stored; wherein if the total size of the total size of the unassigned areas and the unused areas does not exceed the size of the storage areas specified by said area assignment instruction, said management server iteratively performs the steps of determining a storage area, releasing a storage area, and determining whether the total size of the unassigned, the unused areas and the released storage area exceeds the size of the storage areas specified by said area assignment instruction until the total size of the total size of the unassigned areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction or no additional storages areas in which low priority data is stored are available. Naik et al teaches a server assigns priority based on the user who issued the request and keeps information on the mapping (paragraphs 63, 64, and 69).

Shillo and Naik et al are analogous art because they are both related to managing storage on a network.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to use the priority flagging and judging in Naik et al with the system in

Shillo because inefficient use of available resources and high costs are avoided (Naik, paragraph 71).

Shillo in view of Naik et al teaches the limitations as recited above. It fails to teach upon receiving an area assignment instruction, the management server judges whether (i) a size of the unassigned areas exceeds a size of the storage areas specified by said area assignment instruction, (ii) a total size of the unassigned areas and unused areas exceeds the size of the storage areas specified by said area assignment instruction, or (iii) a total size of the unassigned areas, the unused areas and storage areas having stored low-priority data exceeds the size of the storage areas specified by said area assignment instruction, and when the condition (iii) is met, said management server releases at least part of storage areas in which the low-priority data is stored, of the assignment areas of other servers as unassigned areas and assigns at least areas to one of said plurality of servers a code executed when condition (iii) is met, the code including: a code for determining a storage area having a largest number of blocks in which low-priority data is stored, the storage area being of the assignment areas of other servers in which low-priority data is stored wherein the blocks in which low-priority data is stored comprise contiguous blocks, non-contiguous blocks or a combination of non-contiguous and contiguous blocks of low-priority data, a code for releasing as unassigned area the storage area having the largest number of blocks in which low-priority data is stored, a code for determining whether the total size of the unassigned areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction after releasing as unassigned area the storage area having

the largest number of blocks in which low priority data was stored, wherein if the total size of the total size of the unassigned areas and the unused areas does not exceed the size of the storage areas specified by said area assignment instruction, said management server iteratively performs the steps of determining a storage area, releasing a storage area, and determining whether the total size of the unassigned, the unused areas and the released storage area exceeds the size of the storage areas specified by said area assignment instruction until the total size of the total size of the unassigned areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction or no additional storages areas in which low priority data is stored are available. Huntington et al teaches data is distinguished by priority (paragraph 62) and if free space is needed for a request for storage the server may recycle sections that have low priority data within (figure 5, paragraphs 62, 63, and 69).

Shillo in view of Naik et al and Huntington et al are analogous art because they are both related to managing network storage resources.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to use the recycling of low priority data storage in Huntington et al with the system in Shillo in view of Naik et al because large numbers of packets may be communicated and stored on a network with minimal user intervention (Huntington, paragraph 5).

Shillo in view of Naik et al in view of Huntington et al teaches the limitations as recited above. It fails to teach the management server determines a storage area having a largest number of blocks in which low-priority data is stored, the storage area

being of the assignment areas of other servers in which low-priority data is stored; releases as unassigned area the storage area having the largest number of blocks in which low-priority data is stored wherein the blocks in which low-priority data is stored comprise contiguous blocks, non-contiguous blocks or a combination of non-contiguous and contiguous blocks of low-priority data; after releasing the storage area, determines whether the total size of the unassigned areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction; wherein if the total size of the total size of the unassigned areas and the unused areas does not exceed the size of the storage areas specified by said area assignment instruction, said management server iteratively performs the steps of determining a storage area, releasing a storage area, and determining whether the total size of the unassigned, the unused areas and the released storage area exceeds the size of the storage areas specified by said area assignment instruction until the total size of the total size of the unassigned areas and the unused areas exceeds the size of the storage areas specified by said area assignment instruction or no additional storages areas in which low priority data is stored are available. Kurihara et al teaches releasing data of low priority in the cache (column 19, lines 5-19), determining if the cache has free space (column 19, lines 5-19), and releasing data is repeated until it is determined enough space is available (column 19, lines 5-19).

Shillo in view of Naik et al in view of Huntington et al and Kurihara et al are analogous art because they are both related to managing network data storing.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to use the low priority data releasing feature in Kurihara et al with the system in Shillo in view of Naik et al in view of Huntington et al because efficient use of storage areas is provided (Kurihara, column 4, lines 51-58).

Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al teach the limitations as recited above. It fails to teach determining a storage area having a largest number of blocks in which low-priority data is stored. Hazelwood et al teaches the largest block contiguous or non contiguous memory currently available is freed first (section 4.5).

Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al and Hazelwood et al are analogous art because they are both related to managing network data storing.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to use the largest block first release feature in Hazelwood et al with the system in Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al because operation execution speed is increased (Hazelwood, section 1, paragraph 1).

Claim 14 discloses a computer program product according to claim 13, wherein said assignment areas of said storage apparatuses include used areas and unused areas; and said computer program product further comprising: code for information for identifying said used areas and said unused areas of said assignment areas; said code for releasing at least part of assignment areas of other servers as unassigned areas including code for being responsive to the area assignment instruction of storage areas

exceeding unassigned areas received from one of said plurality of servers to release at least part of said unused areas of other servers as unassigned areas on the basis of said identification information. Shillo further teaches the virtual storage pool has used and unused areas (paragraph 43), a server can detect how much allocated space each application actually uses (paragraph 42), and the managing server reallocates the unused portion of the allocated space (paragraph 43).

Claim 16 discloses a computer program product according to claim 14, wherein data stored in said used areas of said storage apparatuses include high-priority data having high priority and low-priority data having low priority; and said computer program product further comprising: code for judging on the basis of a write request of data from one of said plurality of servers whether data to be written in said storage apparatuses is said high-priority data or said low-priority data; and code for information indicative of judgment result and position of storage areas in which said data is written; said code for releasing at least part of unused areas of assignment areas of other servers as unassigned areas including code for being responsive to the area assignment instruction of storage areas exceeding the unassigned areas received from one of said plurality of servers to release at least part of said unused areas and at least part of areas in which said low-priority data is stored, of the assignment areas of other servers as unassigned areas. Shillo further teaches data stored in assigned areas of devices, and reallocating unused resources already allocated (paragraphs 42 and 43). Huntington et al further teaches data is distinguished by priority (paragraph 62) and if

free space is needed for a request for storage the server may recycle sections that have low priority data within (figure 5, paragraphs 62, 63, and 69).

Claims 5, 6, 11, 12, 17, and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shillo (US PGPUB US2003/0110263) in view of Naik et al (US PGPUB US2004/0205206) in view of Huntington et al (US PGPUB US2003/0131098) in view of Kurihara et al (US Patent #6,867,872) in view of Hazelwood et al (Non Patent Literature) as applied to claims 1, 7, and 13 above, and further in view of Karpoff (US PGPUB US2003/0135385).

Claim 5 discloses a management server according to claim 1, wherein said management server makes billing processing for each of said plurality of servers utilizing said storage apparatuses at predetermined intervals. Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al in view of Hazelwood et al teaches the limitations of claim 1 as recited above. It fails to teach of billing each server for the space used at predetermined intervals. Karpoff teaches billing customers based on usage following revenue models similar to the telephone industry, which is widely known to bill a customer on a monthly basis (paragraphs 115 and 116).

Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al in view of Hazelwood et al and Karpoff are analogous art because they are both related to managing storage usage over a network.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to use the billing engine in Karpoff with the system in Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al in view of Hazelwood et al

because a storage service provider is able to charge customers accordingly for standard and convenient features (Karpoff, paragraph 110).

Claim 6 discloses a management server according to claim 5, wherein said management server establishes different billing amounts depending on where low-priority data is stored and high-priority data is stored. Karpoff further teaches billing a customer premiums based on fast access (high priority) or archival (low priority) (paragraphs 99 and 106).

Claim 11 discloses a storage apparatus system according to claim 7, wherein said management server makes billing processing for each of said plurality of servers utilizing said storage apparatuses at predetermined intervals. Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al in view of Hazelwood et al teaches the limitations of claim 7 as recited above. It fails to teach of billing each server for the space used at predetermined intervals. Karpoff teaches billing customers based on usage following revenue models similar to the telephone industry, which is widely known to bill a customer on a monthly basis (paragraphs 115 and 116).

Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al in view of Hazelwood et al and Karpoff are analogous art because they are both related to managing storage usage over a network.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to use the billing engine in Karpoff with the system in Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al in view of Hazelwood et al

because a storage service provider is able to charge customers accordingly for standard and convenient features (Karpoff, paragraph 110).

Claim 12 discloses a storage apparatus system according to claim 11, wherein said management server establishes different billing amounts depending on where low-priority data is stored and high-priority data is stored. Karpoff further teaches billing a customer premiums based on fast access (high priority) or archival (low priority) (paragraphs 99 and 106).

Claim 17 discloses a computer program product according to claim 13, further comprising: code for causing said management server to execute billing processing for each of said plurality of servers utilizing said storage apparatuses at predetermined intervals. Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al in view of Hazelwood et al teaches the limitations of claim 13 as recited above. It fails to teach of billing each server for the space used at predetermined intervals. Karpoff teaches billing customers based on usage following revenue models similar to the telephone industry, which is widely known to bill a customer on a monthly basis (paragraphs 115 and 116).

Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al in view of Hazelwood et al and Karpoff are analogous art because they are both related to managing storage usage over a network.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to use the billing engine in Karpoff with the system in Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al in view of Hazelwood et al

because a storage service provider is able to charge customers accordingly for standard and convenient features (Karpoff, paragraph 110).

Claim 18 discloses a computer program product according to claim 17, further comprising: code for establishing different billing amounts depending on the cases where low-priority data is stored and high-priority data is stored. Karpoff further teaches billing a customer premiums based on fast access (high priority) or archival (low priority) (paragraphs 99 and 106).

Claims 19 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shillo (US PGPUB US2003/0110263) in view of Naik et al (US PGPUB US2004/0205206) in view of Huntington et al (US PGPUB US2003/0131098) in view of Kurihara et al (US Patent #6,867,872) in view of Hazelwood et al (Non Patent Literature) as applied to claim 1 above, and further in view of Honmura et al (US PGPUB US2003/0236790).

Claim 19 discloses a management server according to claim 1, further comprising a storage pool management program, wherein said storage pool management program has at least an unassigned block list and information on a number of unassigned blocks and, when at least said number of unassigned blocks exceeds a size of area assignment requested by said area assignment instruction received from said one of said plurality of servers, determines that the requested area assignment is possible and executes area assignment processing including separating unassigned area of said size from the unassigned block list. Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al in view of Hazelwood et al teaches

the limitations of claim 1 as recited above. It fails to teach of said program has at least an unassigned block list and information on a number of unassigned blocks and, when at least said number of unassigned blocks exceeds a size of area assignment requested by said area assignment instruction received from said one of said plurality of servers, determines that the requested area assignment is possible and executes area assignment processing including separating unassigned area of said size from the unassigned block list. Honmura et al teaches judging based on current information if the requested capacity is available and if the area is available the area is assigned to a device (paragraphs 59, 63, and 64).

Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al in view of Hazelwood et al and Honmura et al are analogous art because they are both related to managing network storage.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to use the judging and assigning features in Honmura et al with the system in Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al in view of Hazelwood et al because a storage service is provided which can easily set the capacity of a client (Honmura, paragraph 17).

Claim 20 discloses a management server according to claim 19, wherein said storage pool management program further comprises storage pool assignment information, said storage pool assignment information including information on a number of unused blocks for each virtual storage area and a server from the plurality of servers to which the virtual storage area is assigned, wherein when a total number of

said number of unassigned blocks and said number of unused blocks exceeds said size of area assignment requested by said area assignment instruction received from said one of said plurality of servers, said storage pool management program determines that the requested area assignment is possible and executes area return processing including issuing an area return instruction to a server to which a virtual storage area having said number of unused blocks has been assigned. Honmura et al further teaches judging based on current information if the requested area is available and if the area is available a server is notified (paragraphs 59, 63, and 65).

Claims 22, 24, and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shillo (US PGPUB US2003/0110263) in view of Naik et al (US PGPUB US2004/0205206) in view of Huntington et al (US PGPUB US2003/0131098) in view of Kurihara et al (US Patent #6,867,872) in view of Hazelwood et al (Non Patent Literature) as applied to claims 1, 7, and 13 above, and further in view of Meier (US Patent #6,295,594).

Claim 22 discloses a management server according to claim 1, wherein in response to an area release instruction, said management server: updates an unassigned block list to include blocks of a storage area identified by the area release instruction; determines a number of used blocks in the storage area identified by the area release instruction, a number of assigned blocks in the storage area, and a number of high-priority blocks in the storage area; decrements a total number of used blocks by the number of used blocks in the storage area to be released; decrements a total number of assigned blocks by the number of assigned blocks in the storage area to

be released; decrements a total number of high-priority blocks by the number of high-priority blocks in the storage area to be released; updates an assignment state bit map, a use state bit map and a data priority bit map, wherein the assignment state bit map, the use state bit map and the data priority bit map comprise a set of bits representing blocks of data in a storage pool, and wherein the assignment state bit map indicates whether each block of data in the storage pool is assigned, the use state bit map indicates whether each block of data in the storage pool is used, and the data priority bit map indicates whether each block of data in the storage pool is high or low priority. Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al in view of Hazelwood et al teaches the limitations of claim 1 as recited above. It fails to teach updates an unassigned block list to include blocks of a storage area identified by the area release instruction; determines a number of used blocks in the storage area identified by the area release instruction, a number of assigned blocks in the storage area, and a number of high-priority blocks in the storage area; decrements a total number of used blocks by the number of used blocks in the storage area to be released; decrements a total number of assigned blocks by the number of assigned blocks in the storage area to be released; decrements a total number of high-priority blocks by the number of high-priority blocks in the storage area to be released; updates an assignment state bit map, a use state bit map and a data priority bit map, wherein the assignment state bit map, the use state bit map and the data priority bit map comprise a set of bits representing blocks of data in a storage pool, and wherein the assignment state bit map indicates whether each block of data in the storage pool is assigned, the

use state bit map indicates whether each block of data in the storage pool is used, and the data priority bit map indicates whether each block of data in the storage pool is high or low priority. Meier teaches a free list is updated in response to de-allocation of data (column 12, lines 10-22), determining the type of data being de-allocated (column 12,lines 10-22), the free list representing the blocks not used is updated to reflect the current availability (column 9, lines 27-49), the allocation cache information representing the assigned blocks is updated to reflect the current availability (column 9, lines 50-62), the reserve list representing the high priority blocks is updated to reflect the current availability (column 10, lines 18-40), and the free list, allocation cache and reserve list are updated to represent the blocks in each category (column 9, lines 27-62 and column 10, lines 18-40).

Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al in view of Hazelwood et al and Meier are analogous art because they are both related to managing storage.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to use the tracking the block status features in Meier with the system in Shillo in view of Naik et al in view of Kurihara et al in view of Hazelwood et al in view of Huntington et al because system performance is improved (Meier, column 3, lines 5-8).

Claim 24 discloses a storage apparatus system according to claim 7, wherein in response to an area release instruction, said management server: updates an unassigned block list to include blocks of a storage area identified by the area release instruction; determines a number of used blocks in the storage area identified by the

area release instruction, a number of assigned blocks in the storage area, and a number of high-priority blocks in the storage area; decrements a total number of used blocks by the number of used blocks in the storage area to be released; decrements a total number of assigned blocks by the number of assigned blocks in the storage area to be released; decrements a total number of high-priority blocks by the number of high-priority blocks in the storage area to be released; updates an assignment state bit map, a use state bit map and a data priority bit map, wherein the assignment state bit map, the use state bit map and the data priority bit map comprise a set of bits representing blocks of data in a storage pool, and wherein the assignment state bit map indicates whether each block of data in the storage pool is assigned, the use state bit map indicates whether each block of data in the storage pool is used, and the data priority bit map indicates whether each block of data in the storage pool is high or low priority. Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al in view of Hazelwood et al teaches the limitations of claim 7 as recited above. It fails to teach updates an unassigned block list to include blocks of a storage area identified by the area release instruction; determines a number of used blocks in the storage area identified by the area release instruction, a number of assigned blocks in the storage area, and a number of high-priority blocks in the storage area; decrements a total number of used blocks by the number of used blocks in the storage area to be released; decrements a total number of assigned blocks by the number of assigned blocks in the storage area to be released; decrements a total number of high-priority blocks by the number of high-priority blocks in the storage area to be released; updates an

assignment state bit map, a use state bit map and a data priority bit map, wherein the assignment state bit map, the use state bit map and the data priority bit map comprise a set of bits representing blocks of data in a storage pool, and wherein the assignment state bit map indicates whether each block of data in the storage pool is assigned, the use state bit map indicates whether each block of data in the storage pool is used, and the data priority bit map indicates whether each block of data in the storage pool is high or low priority. Meier teaches a free list is updated in response to de-allocation of data (column 12, lines 10-22), determining the type of data being de-allocated (column 12,lines 10-22), the free list representing the blocks not used is updated to reflect the current availability (column 9, lines 27-49), the allocation cache information representing the assigned blocks is updated to reflect the current availability (column 9, lines 50-62), the reserve list representing the high priority blocks is updated to reflect the current availability (column 10, lines 18-40), and the free list, allocation cache and reserve list are updated to represent the blocks in each category (column 9, lines 27-62 and column 10, lines 18-40).

Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al in view of Hazelwood et al and Meier are analogous art because they are both related to managing storage.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to use the tracking the block status features in Meier with the system in Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al in view of Hazelwood et al because system performance is improved (Meier, column 3, lines 5-8).

Claim 26 discloses a computer program product to claim 13, wherein in response to an area release instruction, said management server: updates an unassigned block list to include blocks of a storage area identified by the area release instruction; determines a number of used blocks in the storage area identified by the area release instruction, a number of assigned blocks in the storage area, and a number of high-priority blocks in the storage area; decrements a total number of used blocks by the number of used blocks in the storage area to be released; decrements a total number of assigned blocks by the number of assigned blocks in the storage area to be released; decrements a total number of high-priority blocks by the number of high-priority blocks in the storage area to be released; updates an assignment state bit map, a use state bit map and a data priority bit map, wherein the assignment state bit map, the use state bit map and the data priority bit map comprise a set of bits representing blocks of data in a storage pool, and wherein the assignment state bit map indicates whether each block of data in the storage pool is assigned, the use state bit map indicates whether each block of data in the storage pool is used, and the data priority bit map indicates whether each block of data in the storage pool is high or low priority. Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al in view of Hazelwood et al teaches the limitations of claim 13 as recited above. It fails to teach updates an unassigned block list to include blocks of a storage area identified by the area release instruction; determines a number of used blocks in the storage area identified by the area release instruction, a number of assigned blocks in the storage area, and a number of high-priority blocks in the storage area; decrements a total number of used blocks by the

number of used blocks in the storage area to be released; decrements a total number of assigned blocks by the number of assigned blocks in the storage area to be released; decrements a total number of high-priority blocks by the number of high-priority blocks in the storage area to be released; updates an assignment state bit map, a use state bit map and a data priority bit map, wherein the assignment state bit map, the use state bit map and the data priority bit map comprise a set of bits representing blocks of data in a storage pool, and wherein the assignment state bit map indicates whether each block of data in the storage pool is assigned, the use state bit map indicates whether each block of data in the storage pool is used, and the data priority bit map indicates whether each block of data in the storage pool is high or low priority. Meier teaches a free list is updated in response to de-allocation of data (column 12, lines 10-22), determining the type of data being de-allocated (column 12,lines 10-22), the free list representing the blocks not used is updated to reflect the current availability (column 9, lines 27-49), the allocation cache information representing the assigned blocks is updated to reflect the current availability (column 9, lines 50-62), the reserve list representing the high priority blocks is updated to reflect the current availability (column 10, lines 18-40), and the free list, allocation cache and reserve list are updated to represent the blocks in each category (column 9, lines 27-62 and column 10, lines 18-40).

Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al in view of Hazelwood et al and Meier are analogous art because they are both related to managing storage.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to use the tracking the block status features in Meier with the system in Shillo in view of Naik et al in view of Huntington et al in view of Kurihara et al in view of Hazelwood et al because system performance is improved (Meier, column 3, lines 5-8).

Response to Arguments

Applicant's arguments filed March 11, 2008 have been fully considered but they are not persuasive.

Applicant asserts the prior art fails to disclose or suggest determining a storage area having a largest number of blocks in which low-priority data is stored, and iteratively performing the steps of determining a storage area, releasing the storage area and determining whether the total size of the unassigned, the unused areas and the released storage area is at least the size of the storage areas specified by said area assignment instruction until the total size of the total size of the unassigned areas and the unused areas is at least the size of the storage areas specified by said area assignment instruction. The Examiner respectfully disagrees, Kurihara et al teaches selectively deleting low priority data by a predetermined method until enough space is available (column 19, lines 5-19) and Hazelwood et al teaches the method of selecting and releasing the largest element of contiguous or non contiguous memory first (section 4.5).

In response to applicant's argument that Kurihara et al and Hazelwood et al are nonanalogous art, it has been held that a prior art reference must either be in the field of applicant's endeavor or, if not, then be reasonably pertinent to the particular problem

with which the applicant was concerned, in order to be relied upon as a basis for rejection of the claimed invention. See *In re Oetiker*, 977 F.2d 1443, 24 USPQ2d 1443 (Fed. Cir. 1992). In this case, Kurihara and Hazelwood et al are both related to processing stored data.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Brian J. Gillis whose telephone number is (571)272-7952. The examiner can normally be reached on M-F 7:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Rupal Dharia can be reached on 571-272-3880. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Brian J Gillis
Examiner
Art Unit 2141

/B. J. G./
Examiner, Art Unit 2141
6/20/2008

/Jason D Cardone/
Supervisory Patent Examiner, Art Unit 2145